



King's Research Portal

DOI:

[10.1007/s10458-015-9316-2](https://doi.org/10.1007/s10458-015-9316-2)

Document Version

Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Haberland, V., Miles, S., & Luck, M. (2017). Negotiation strategy for continuous long-term tasks in a grid environment. *Autonomous Agents and Multi-Agent Systems*. <https://doi.org/10.1007/s10458-015-9316-2>

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Negotiation Strategy for Continuous Long-Term Tasks in a Grid Environment

Valeriia Haberland · Simon Miles · Michael Luck

Received: date / Accepted: date

Abstract Nowadays, much research is concerned with execution of long-term continuous tasks, which produce data in real time, e.g. monitoring applications. These tasks can be run for months or years and they are usually resource intensive in terms of the large amounts of data which is processed per time unit. A Grid can potentially provide the amount of resources necessary to execute these tasks, but it might prove to be impossible or non-beneficial for a Grid to allocate resources for such long durations as these resources can be also requested by other clients or might join a Grid only for some periods of time. To resolve these differences, a client and a Grid Resource Allocator negotiate, and a client has to agree for a shorter execution period at the end of which it needs to negotiate again. In this paper, we discuss in detail a decision-making mechanism for a client as part of its negotiation strategy, which aims to increase the duration of execution periods and to decrease the duration of interruptions. This new strategy, ConTask, has been tested on a realistic Grid resource simulator, and it demonstrates better utilities than our strategy which has not been specifically designed for continuous tasks under various conditions.

Keywords Automated negotiation · Continuous task · Grid resources

1 Introduction

Many researchers focus on processing data streams, which are obtained from sensors. For example, these data streams may monitor the level of air pollution [13], or the possibility of earthquake [10]. The tasks which process these data streams produce data in real time and, therefore, it is desirable for these tasks to be run continuously, but short interruptions might be acceptable, i.e. they do not noticeably affect the success of task execution. A duration of interruption is considered to be acceptable if significant changes are unlikely to occur during this time, e.g. a substantial rise or drop in the temperature, leading to a loss of relevant information for a client. It has to be noted that a client knowledge of the observed parameter (e.g. temperature) becomes less realistic with time if task is interrupted, causing the client utility to be decreased at higher rates. These tasks are often required to run for unspecified or long periods of time, which can be months or years. In our work, we consider that these

tasks are intended to run for specified long periods of time rather than uncertain periods of time and they are considered to be *continuous long-term tasks*.

A continuous task, as discussed above, is appropriate to execute in a Grid [12], because it potentially comprises more resources (e.g. CPUs) than any cluster or supercomputer. A Grid generally follows a concept of non-commercial resource sharing¹, which means that resources can be allocated with no monetary cost for a client. However, a Grid can be accessible by any clients who wish to run their tasks and, therefore, the resource availability may vary over time. Moreover, some resources may leave or join a Grid in future. In our work, we assume that resource availability changes *near-periodically* over time mostly due to the pseudo-periodicity in demand on resources. This assumption is realistic for some Grids [4, 20], e.g. Iosup et al. [20] show some periodicity patterns in their illustration of Grid resource utilisation as well as Andrzejak and Ceyran [4] demonstrate a periodicity in CPU utilisation for some servers. For instance, Kondo et al. [21] discuss CPU utilisation in desktop grids, which is generally larger during weekdays than weekends. Their study also suggests that CPUs are on average unavailable 19% of the time during working days and only 3% of the time during weekends, where availability means that a CPU should have more than 1% of free capacity to perform a task. Hence, relative predictability of the pseudo-deterministic changes in resource availability is utilised in our work to support near-continuous tasks execution.

We assume that it is impossible or non-beneficial for a Grid to allocate resources for the whole requested long durations of time, because long-term predictions of resource availability are generally unreliable for highly dynamic and open environments [3, 38] and these resources can be also requested by other clients. In other words, a Grid might be unable to guarantee task execution in the long-term future and, hence, unwilling to commit to its execution. Therefore, a task is likely to be granted a shorter period of resource utilisation than initially requested. Then, this task is interrupted at the end of this period and this interruption is considered to be *planned*. If a task has been interrupted, it has to request resources from a Grid again. Frequently, negotiation is used to agree the requirements between clients' and resource providers' [30]. In our work, a negotiation is bilateral and it is initiated by a client immediately after task's interruption. A resource provider's interests are represented by a *Grid Resource Allocator* (GRA) and a client's interests by a *Client Agent*, where both are autonomous agents [37]. The abilities of autonomous agents to monitor continuously the environment and respond actively to the dynamic changes in this environment are important for automated negotiation in such open environments as a Grid. The other relevant agent's ability is to communicate and decide on their actions without human commands, which becomes crucial for the huge number of simultaneously negotiating agents.

In this paper, we present a client's negotiation strategy, *ConTask*, which allows a client to concede beyond the last best proposal of the GRA or its own proposal in order to start the subsequent interruption period under favourable conditions for negotiation, i.e. closer to the maximum of resource availability. This maximum is estimated to some degree, considering pseudo-periodicity in resource availability fluctuations. It should be noted that the client is unaware of resource availability fluctuations and estimates them based on the outcome of negotiation, i.e. the agreed duration of task execution. The ConTask strategy also offers a largely extended *evaluation function*, compared to [16], which is used to decide whether the current and / or cumulative (all previous interruptions) durations of interruption are too long, while the previous strategy [16] only considers a risk of resource exhaustion before the end of negotiation. If at least one of those interruptions is too long, a client generally tends to be

¹ We do not focus on commercial Grids.

more generous in negotiation. This extended evaluation function leads to considerably better utilities for a client than its previous version, which takes into account only the risk of resource exhaustion during negotiation. The paper is structured as follows. Section 2 discusses related work, while Section 3 presents a formal model for a continuity of task execution and negotiation for resources. Section 4 describes our ConTask negotiation strategy and Section 5 discusses the evaluation results. Finally, Section 6 concludes this paper.

2 Background

Research and industry currently face the issues of continuous or near-continuous task execution on plants or in robotics, air pollution monitoring, etc. For example, Castro et al. [9] aim to find an optimal schedule of resource usage (e.g. equipment) on plants for continuous processes (e.g. continuous feedstock mixing). Other research [2] addresses the issues of continuous area monitoring by a group of robots in order to detect some events of interest (e.g. gas leaks). The work of Chen et al. [10] proposes a Grid-based Adaptive Execution on Streams (GATES) to process distributed data streams in a real-time manner. For instance, this Grid system can be used to process tasks, which can predict large earthquakes based on ground movements. Although all these researchers offer effective mathematical or algorithmic solutions for their problem domains, they mostly focus on matching or scheduling resource-task pairs. Hence, a client (an owner of a task) cannot effectively influence task execution or affect possible interruptions of tasks.

A client can significantly influence task execution when its requirements can be negotiated with the GRA, where both parties can utilise various strategies in order to follow their objectives. Strategies which depend on time naturally reflect many real-life time-dependent tasks. Many such strategies have been developed in much work [1, 7, 11, 24, 31], where an amount of concession changes over time, considering usually an approaching negotiation deadline. If this deadline is reached without agreement, then a negotiation terminates.

For example, Faratin et al. [11] propose several time-dependent strategies, where a negotiator significantly concedes closer to its negotiation deadline (greedy behaviour) or it significantly concedes from the beginning of negotiation (generous behaviour). Lang [22] also mentions a strategy when a negotiator concedes linearly over time, while Sim [29] introduces a “waiting” strategy, where a negotiator concedes only at the deadline. Such work not only focuses on time in their strategies, but also takes into account changing market competition and opportunity (e.g. [1, 30, 31]), the behaviour of the negotiator’s opponent (e.g. [1, 11, 18, 28]), and even strategies that do not depend on time [7].

In particular, Sim [30] describes a *Market-Driven Agent* (MDA) which considers time and market-related pressure on a negotiator. In this strategy, the opportunity for a negotiator is determined by the number of potential trading partners and the differences between their and negotiator’s proposals in terms of their utilities. The competition pressure is determined by the probability that a negotiator is chosen as the most preferable partner by at least one of its trading partners in a particular negotiation round, and the pressure potentially increases with the number of competitors. The time-related pressure on the MDA is similar to the time-dependent tactics, presented by Faratin et al. [11], where a pressure on a negotiator increases when a negotiation approaches the deadline, and a negotiator can adopt generous, indifferent (linear) or greedy behaviour. Sim et al. [32, 33] also consider that a negotiator is able to decrease its utility expectation when a pressure becomes high by accepting a ‘slightly’ worse proposal from its opponent, where a quantitative estimate of ‘slightly’ or ‘not slightly’ is determined through a fuzzy logic mechanism. Although this work allows a

negotiator to accept a smaller than expected utility, it does not take into account an effect of such decision for future negotiations, which is essential for continuous long-term task execution.

The MDA agent [30] has been extended in many other research papers. For instance, Ren et al. [26] re-calculate the opportunity and competition factors by considering possible future market changes, e.g. the probability of n trading partners joining the negotiation process in the next round. Adabi et al. [1] state that it is important to consider a past concession behaviour of the trading partners in order to punish or encourage an unfair or favourable behaviour respectively. However, Adabi et al. assume that a negotiator (e.g. client) has an access to the database which stores a concession history of its opponents, which may not be available to a client. There are other negotiation strategies which consider an opponent's behaviour [11, 18, 19], and they range from imitating this behaviour [11] to learning the opponent's tactics [19] or preference profile [18], etc. However, those strategies do not focus on learning the causes behind the opponent's behavioural changes such as environmental changes.

It has to be noted that if the end of negotiation is *unspecified* (no specific deadline), but the duration of negotiation is relevant for client's tasks, especially for continuous tasks, then the mentioned above strategies do not account to this issue. They also do not consider the cumulative effect of previous interruptions on the current negotiation, e.g. a more generous client if a cumulative interruption is too long.

Some researchers focus on developing and improving a platform *Generic Environment for Negotiation with Intelligent multi-purpose Usage Simulation* (GENIUS) [5, 6, 17] for multi-issue negotiation simulation between heterogeneous agents. It has a user interface, which allows a human user to choose the domain of negotiation, the negotiating agents with their strategies, preferences, etc. This platform also allows integration of new negotiation agents and it can compare an outcome of negotiation with some optimal outcomes such as a Pareto efficient outcome. This platform is also used to perform tournaments among agents, where the agents' performance can be contested in the different scenarios. However, this platform is not designed to simulate negotiation processes for continuous task execution, where future negotiations (e.g. a client's tactic) depend on the duration and success of previous negotiations, e.g. a too long cumulative interruption leads to more generous client's tactic.

It is generally desirable for a client to ensure that an agreement can be reached as fast and with as good an outcome as possible. This can be achieved if a demand on resources is not high and resources are genuinely available. Hence, it is beneficial for a client to find out when resource availability is at its maximum, assuming pseudo-periodicity in workload and resource availability over time, and to negotiate during this period. Much work [8, 25, 34, 36, 38] focuses on predictions of Grid performance to improve scheduling of tasks, but also usually emphasise the role of the GRA rather than client, which may not have direct knowledge of resource availability in a Grid. For example, Wolski et al. [35, 36] describe the *Network Weather Service* which produces the short-term predictions of meta-computing resource performance, considering its network and computational characteristics, in a distributed environment based on historical data. Nudd et al. [25] propose a toolset, the *Performance Analysis and Characterisation Environment*, which allows application developers to predict and evaluate a performance of sophisticated applications (e.g. parallel applications) at the different stages of their development and deployment. However, these performance predictions involve a knowledge of performance characteristics of available resources, which might not be accessible by a client.

3 Formal model

In our work, we assume that each task can be interrupted one or more times during its long-term execution. If a task has been interrupted, it must again request resources from a Grid. However, if resources are scarce and demand on resources is high at the time of interruption, then a client may keep negotiating with the GRA much longer than if resources are available, which may significantly decrease client utility. This section provides a formal model for continuous execution of a task, including a client's utility, as well as negotiation itself. Note that this formal model of continuous task execution has been presented in the work of Haberland et al. [15].

3.1 Continuous task model

The start time of task execution t^{str} is when a client has submitted its first request for resources to a Grid, while the end moment for the whole long-term duration of task execution is t^{end} . We assume that a client may have multiple tasks, but they all run independently. In order to compare and average utilities over multiple tasks, we assume that each task has the same global start t^{str} and end t^{end} time. The total execution time for each task is $\tau^{tot} = t^{end} - t^{str}$.

Each task i may have several interruption(s) within τ^{tot} , which are not simultaneous for multiple tasks. Each adjoined pair $(\tau_{i,l}^{int}, \tau_{i,l}^{exe})$ of a single interruption $\tau_{i,l}^{int}$ and execution $\tau_{i,l}^{exe}$ periods of time for continuous task i , where an interruption period precedes a corresponding execution period, has its counter $l = 1, 2, 3, \dots$ within τ^{tot} . Here, each $\tau_{i,l}^{exe}$ starts at $t_{i,l}^{str}$ and ends at $t_{i,l}^{end}$. Task interruption or execution at time t (i.e. when in progress) is $\tau_{i,l}^{int}(t)$ or $\tau_{i,l}^{exe}(t)$ respectively, where $t \in [t^{str}, t^{end}]$.

We assume that a cumulative duration of interruption over time objectively reflects on the success of the client's strategy, because each interruption period might have lighter or tougher conditions for negotiation (e.g. level of resource availability). We define a cumulative duration of interruption $\tau_{i,l}^{cum} = \sum_{k=1}^l \tau_{i,k}^{int}$ as the sum of all interruption durations from the first until the l^{th} prior to the execution period $\tau_{i,l}^{exe}$. A cumulative interruption of a task at time t , when a task is inside an interruption period $\tau_{i,l}^{int}(t)$, is $\tau_{i,l}^{cum}(t)$, where $t \in [t^{str}, t^{end}]$.

Client utility considers all interruption and execution periods for each task, where only execution periods explicitly contribute to utility. We argue that the utility gained by a client for each execution period $\tau_{i,l}^{exe}$ must be affected negatively to a varying degree by two factors: the duration of the preceding interruption $\tau_{i,l}^{int}$, which shows how much successfully a client negotiated for this execution period; the cumulative duration of interruption $\tau_{i,l}^{cum}$, which shows how much successfully a client generally negotiated for previous execution periods, including $\tau_{i,l}^{exe}$. To derive client utility, we introduce an *effectiveness function* $E_{i,l}(t)$ which shows the success level of execution of task i as a process over time within each execution period $\tau_{i,l}^{exe}(t)$, assigning a value from the range $[0, 1]$ to each point in time t . Figure 1 shows the effectiveness of task execution over time as alternating execution and interruption periods. The effectiveness function increases linearly when the task is running (during $\tau_{i,l}^{exe}(t)$) and does not change when the task is interrupted (during $\tau_{i,l}^{int}(t)$). The length of interruptions (single and cumulative) affects the values of the effectiveness function during the following execution period (e.g. $E_{i,l=2}(t_{i,l=2}^{str}) < E_{i,l=1}(t_{i,l=1}^{end})$).

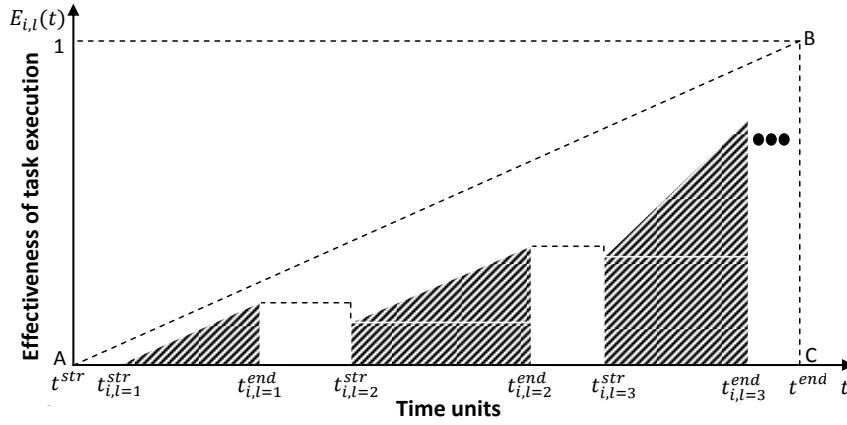


Fig. 1 An effectiveness of continuous task execution over time

The level of impact of the length of interruption on the following execution period is calculated with damping functions $SI(\tau_{i,l}^{int})$ and $CI(\tau_{i,l}^{cum})$ for a single and cumulative interruptions respectively. These functions have a form of broadened step functions and they produce values from the interval $[0, 1]$ where 1 denotes no impact. The values of these functions decrease during the interruption period from 1 towards 0. Each of these functions has two variables which define how fast the possible increment of the client utility for a particular execution period decreases towards zero, depending on the duration of interruption period. They are $(\tau_{int}^{max}, \epsilon_{int})$ for a single interruption, while $(\tau_{cum}^{max}, \epsilon_{cum})$ for a cumulative interruption. Specifically, the variables τ_{int}^{max} or τ_{cum}^{max} denote the durations of single or cumulative interruptions after which the corresponding functions' values decrease by half compared to their maximal value, which is considered to be a substantial level of impact for a client. The variables ϵ_{int} or ϵ_{cum} determine the speed of decrease in the corresponding increment of the client's utility, caused by interruption, and this speed increases closer to the end of period τ_{int}^{max} or τ_{cum}^{max} . For example, a continuous task which monitors temperature in the building has been interrupted. The temperature will not rapidly change, but our knowledge about its value becomes more and more unrealistic when time passes, and therefore it should more significantly affect the effectiveness of task execution. These functions are presented below:

$$SI(\tau_{i,l}^{int}) = \frac{1}{e^{(\tau_{i,l}^{int} - \tau_{int}^{max})/\epsilon_{int}} + 1}. \quad (1)$$

$$CI(\tau_{i,l}^{cum}) = \frac{1}{e^{(\tau_{i,l}^{cum} - \tau_{cum}^{max})/\epsilon_{cum}} + 1}. \quad (2)$$

Here, if $\tau_{i,l}^{int} \ll \tau_{int}^{max}$, then the value of $SI(\tau_{i,l}^{int})$ is close to one. If $\tau_{i,l}^{int} \gg \tau_{int}^{max}$, then the value of $SI(\tau_{i,l}^{int})$ is close to zero. The smaller the value of $SI(\tau_{i,l}^{int})$ or $CI(\tau_{i,l}^{cum})$, the less effective task execution becomes in future.

The effectiveness function $E_{i,l}(t)$ increases in the range from the level of effectiveness $E_{i,l-1}(t_{i,l-1}^{end})$, achieved by a task before interruption $\tau_{i,l}^{int}$ at time $t_{i,l-1}^{end}$, multiplied by the values of functions $SI(\tau_{i,l}^{int})$ and $CI(\tau_{i,l}^{cum})$, towards the largest possible effectiveness 1

at time t^{end} in the time interval $\tau_{i,l}^{exe}$, as below if $j \equiv l - 1$:

$$E_{i,l}(t) = \begin{cases} \left[\frac{(1 - E_{i,j}(t_{i,j}^{end}))t + E_{i,j}(t_{i,j}^{end})t^{end} - t_{i,l}^{str}}{t^{end} - t_{i,l}^{str}} \right] \times \\ SI(\tau_{i,l}^{int}) \times CI(\tau_{i,l}^{cum}) & , \text{ if } t \in \tau_{i,l}^{exe}, \\ E_{i,j}(t_{i,j}^{end}) & , \text{ if } t \in \tau_{i,l}^{int}. \end{cases} \quad (3)$$

Here, if $t = t_{i,l}^{str}$, then the effectiveness of task execution $E_{i,l}(t) = E_{i,j}(t_{i,j}^{end}) \times SI(\tau_{i,l}^{int}) \times CI(\tau_{i,l}^{cum})$, which means that the effectiveness function at the beginning of execution period $\tau_{i,l}^{exe}$ corresponds to its value at the end of previous execution $\tau_{i,j}^{exe}$, and it is affected by the durations of interruptions. If $t = t^{end}$, then the effectiveness of task execution $E_{i,l}(t) = SI(\tau_{i,l}^{int}) \times CI(\tau_{i,l}^{cum})$. Consequently, if task execution reaches its deadline t^{end} with interruptions, then the effectiveness of task execution will not be equal to its largest possible value, i.e. 1.

Client utility U_i for a task i is the sum of all areas under the effectiveness function for the execution periods in proportion to the maximum possible area when the task is run without interruptions during τ^{tot} , and is shown in Figure 1 as the area of 'ABC'. This utility is presented in Equation (4).

$$U_i = \frac{1}{\tau^{tot}} \sum_{l=1}^{L_i} \left(E_{i,l}(t_{i,l}^{str}) + E_{i,l}(t_{i,l}^{end}) \right) \times \tau_{i,l}^{exe}, \quad (4)$$

where L_i is the total number of interruption-execution pairs within τ^{tot} for task i . The average utility $U_{aver} = 1/N \sum_{i=1}^N U_i$ is calculated as an average of utilities for N tasks, and used in our evaluation.

3.2 Negotiation model

In this work, a polynomial function, presented by Faratin et al. [11] has been chosen as a basic negotiation strategy for a client, because it allows a client to concede faster at the beginning of negotiation as compared to an exponential function for conceding tactics. This distinction between a polynomial and exponential functions is crucial for a client, because a client becomes generous in negotiation when there is a risk of resource exhaustion. In the cases of client's greedy behaviour, a polynomial function produces much smaller values at the beginning of negotiation than an exponential function which would preserve the client utility, and this function also allows a client to wait shorter time before start conceding substantially which might prevent negotiation failure. While the work of Faratin et al. tested different time-dependent strategies, the number of possible tactics in their experimental settings was finite, which limits a choice for a client. Our ConTask negotiation strategy not only assumes the infinite number of tactics, but the choice of those tactics is determined through a sophisticated mechanism at every negotiation round, which takes into consideration task-specific characteristics (e.g. continuity) and dynamics of Grid environment as discussed further in this paper.

A client and GRA negotiate using an adopted *alternating proposals protocol* [27], where the parties send proposals in turns, and both can accept the opponent's proposal or quit a negotiation process. In our work, a client has to negotiate with the GRA until resources are

obtained which leads to uncertainty in respect of the final deadline of negotiation. However, a negotiation process is time-dependent in terms of the length of interruption. Therefore, we utilise time-dependent strategies [11] with reasonably short deadlines by the end of which a client should not lose significant utility, and a client repeats these *short negotiations* until an agreement is reached. In particular, short negotiations are defined in our experimental settings further in this paper. In an ideal scenario, a client should be able to reach agreement during a single short negotiation. Agreement is reached when a client accepts the GRA's proposal, or the GRA accepts a client's proposal after it has been confirmed as the final one by a client.

In each short negotiation, a negotiator concedes towards its opponent, starting from an optimal length of task execution² $\tau_{i,l}^{COp}$ for a client and a minimum length $\tau_{i,k}^{GMn}$ for the GRA, where the client's optimal value can change at the beginning of each interruption period, but the GRA's minimum can change at the beginning of each short negotiation (k is the counter of short negotiations). A negotiation proceeds towards the negotiators' *reservation values*, i.e. the minimum acceptable length of task execution $\tau_{i,l}^{CMn}$ for a client and the maximum available length $\tau_{i,r}^{GMx}$ for the GRA, which can change every next r negotiation round. The GRA varies its reservation value in order to reflect on the changes in resource availability, e.g. if resources become less available, its reservation value decreases. A short negotiation stops when the negotiators reach agreement, one of them quits (e.g. the GRA if resources are exhausted) or the short negotiation deadline t_{dl} is reached. The level of concession depends on the *greediness* $\beta_{i,r}^C$ for the client and $\beta_{i,r}^G$ for the GRA. If a proposal from a client is $\tau_{i,r}^C$ for task i at round r , and the proposal from the GRA is $\tau_{i,r}^G$, then:

$$\tau_{i,r}^C = \tau_{i,l}^{COp} + \left(\frac{r \times \Delta t}{t_{dl}} \right)^{\beta_{i,r}^C} \times (\tau_{i,l}^{CMn} - \tau_{i,l}^{COp}), \quad (5)$$

$$\tau_{i,r}^G = \tau_{i,k}^{GMn} + \left(\frac{r \times \Delta t}{t_{dl}} \right)^{\beta_{i,r}^G} \times (\tau_{i,r}^{GMx} - \tau_{i,k}^{GMn}), \quad (6)$$

where Δt is equal to one conventional time unit. Note that $\tau_{i,r}^{GMx}$ increases or decreases proportionally to the level of resource availability. In these equations, when $r = 0$, then $\tau_{i,r}^C = \tau_{i,l}^{COp}$ and $\tau_{i,r}^G = \tau_{i,k}^{GMn}$. When $r \times \Delta t = t_{dl}$, then $\tau_{i,r}^C = \tau_{i,l}^{CMn}$ and $\tau_{i,r}^G = \tau_{i,r}^{GMx}$. The level of greediness for both negotiators changes, according to three tactics [11, 22]: *greedy* ($\beta_{i,r}^{C,G} > 1$), *generous* ($0 < \beta_{i,r}^{C,G} < 1$) or *indifferent* ($\beta_{i,r}^{C,G} = 1$).

The GRA's level of greediness is based on the change in its reservation value and the demand on resources, e.g. if demand is high and reservation value decreases, then it becomes more greedy. The client's level of greediness is estimated according to our extended evaluation function, discussed in this paper, which controls a composition of fuzzy sets [14, 16]. A *fuzzy set* A is a subset of Y where each element $y \in Y$ is mapped to the value from the interval $[0, 1]$, which determines the degree of membership of y in A , defined by the membership function $\mu_A(y)$ [23, 39]. Each fuzzy set has a range of values of $y \in Y$ which have a non-zero membership. For example, a decrease in the level of greediness for a client between 50% and 100% of its previous value can be considered as a substantial one with the different or same levels of certainty for the different values from this range. In this work, those ranges for the number of fuzzy sets are determined based on the extended evaluation function, presented in this paper.

² Note, "C" denotes a client and "G" denotes the GRA, while "Op", "Mn" and "Mx" denote an optimal, minimum and maximum length of time in the upper index of the notations, presented in this section.

This evaluation function shows the level of attractiveness of a specific level of greediness, produced by a fuzzy logic mechanism, for a client at each negotiation round, considering different factors such as the risk of resource exhaustion. The ranges of fuzzy sets are varied and their combinations constitute different compositions of fuzzy sets, which produce different levels of greediness for a client. In this paper, a client also chooses to concede beyond the best last proposal from the GRA (or beyond its own non-confirmed last proposal accepted by the GRA) in a short negotiation to start the next interruption period at approximately maximum resource availability.

4 Negotiation strategy

The goal of our ConTask negotiation strategy for a client is to reduce the length of interruption periods and increase the length of execution periods, while not losing significantly in utility³. As long as resource availability is assumed to change pseudo-periodically [4, 20], then the reservation value of the GRA fluctuates periodically as well. A client is unaware of resource availability or demand on resources, but higher demand generally leads to more scarce resources. Resource scarcity leads to a less generous GRA with a smaller reservation value. As a result, it is likely that a task is allocated a shorter execution period when resources are scarce. Hence, a client can determine which time interval is *unfavourable* for negotiation and which is not based on the durations of agreed execution periods. That is, the longer an allocated execution period is as a result of negotiation, the more favourable this negotiation interval. Therefore, a client can estimate when the resource availability will be at maximum, and aim to negotiate during this time (see Subsection 4.1). A client estimates it through preliminary short negotiations with the GRA within some period before t^{str} , which is not noticeable to the GRA as those negotiations follow the same rules as any negotiations after t^{str} . Here, the accuracy of locating this maximum depends on the length of this period, the amount of statistics and / or the level of determinism in the GRA's reservation value fluctuations (i.e. those fluctuations might have large or smaller oscillations over time). Therefore, a client may need another technique, except negotiating at the maximum resource availability, in order to reduce the durations of interruptions. We offer an extended evaluation function, which allows a client to become more generous in negotiation when single or cumulative interruptions are too long (see Subsection 4.2).

4.1 Negotiating at maximum resource availability

A client generally aims to obtain a longer execution period during negotiation. Conventionally, an agreement is reached when the negotiator's utility for the opponent's offer is higher than for the next possible negotiator's offer. In our work, a client does not always accept a proposal which leads to a better utility than its own next proposal, but may decide to concede more by proposing a shorter execution period to the GRA if this shortened period ends closer to the maximum resource availability. This decision is mostly connected with the *continuity* of task execution, because a client tries not only to obtain a better outcome in the current negotiation, but must improve the overall effectiveness of task execution by reducing all possible interruptions within τ^{tot} , i.e. one successful negotiation affects any following one in terms of the *initial conditions* such as resource availability, etc.

³ Note, the ideas of this strategy have been very briefly discussed in a short paper [15].

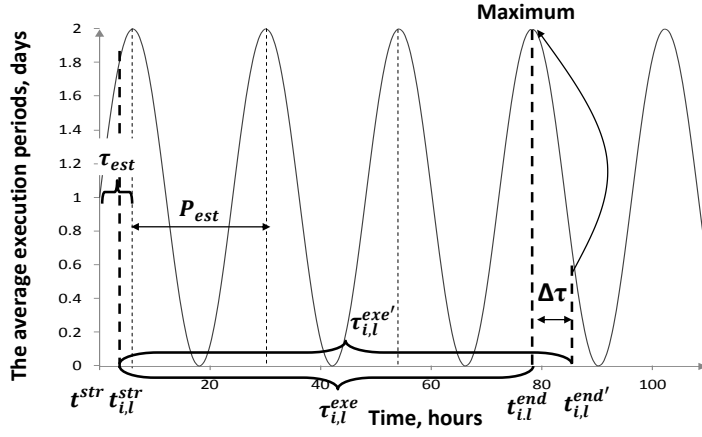


Fig. 2 A decision to shorten an execution period $\tau_{i,l}^{exe'}$

Here, we define the *best offered execution period* $\tau_{i,l}^{exe'}$ as the last proposal of the GRA or a client, which is to be accepted by a client or which is already accepted by the GRA (not confirmed by a client) [15]. That is, it is a proposal that should conventionally seal an agreement between negotiators. However, a client may decide to ask for a shorter execution period $\tau_{i,l}^{exe}$ in order for this period to end around the maximum of resource availability. It also has to be noted that a client attempts to reduce the length of $\tau_{i,l}^{exe'}$ only as much as to reach the closest maximum from the end of this execution period.

Assume the length of time on which a client shortens its execution period is $\Delta\tau$. The expected end time of $\tau_{i,l}^{exe'}$ is $t_{i,l}^{end'}$. The shortened execution period $\tau_{i,l}^{exe}$ must end around the maximum resource availability as in Figure 2⁴. A client tries to choose as short as possible $\Delta\tau$ to lose as little utility as possible. Assume that an estimated period of a pseudo-periodic resource availability fluctuations is P_{est} , while an estimated time interval between t^{str} and the first maximum of resource availability is τ_{est} . Both values are derived from accumulated statistics of average execution periods per time interval (e.g. per half hour) during preliminary short negotiations. Note that if the execution period $\tau_{i,l}^{exe'}$ does not contain at least one maximum of resource availability, a client does not shorten it because $\tau_{i,l}^{exe} < 0$. If the end of $\tau_{i,l}^{exe'}$ is already in the maximum, $\Delta\tau = 0$. That is, $\Delta\tau$ is calculated as follows:

$$\Delta\tau = (t_{i,l}^{end'} - t^{str}) - (\tau_{est} + P_{est} \times n), \quad (7)$$

where n is the number of the corresponding periods P_{est} until the estimated maximum resource availability, where the execution period would be expected to end if it becomes shorter. Consequently, the shortened execution period is $\tau_{i,l}^{exe} = \tau_{i,l}^{exe'} - \Delta\tau$, while its expected end time of $\tau_{i,l}^{exe}$ is $t_{i,l}^{end}$. The algorithm to shorten the best offered execution period is presented below.

⁴ The figure shows a strongly periodic dependence for average execution periods over time only as an example.

Algorithm 1 Client algorithm to shorten $\tau_{i,l}^{exe'}$

```

1: {One negotiator offers  $\tau_{i,l}^{exe'}$ , but there is no agreement yet}
2: Calculate  $\Delta\tau$  (see Formula (7))
3: if  $\Delta\tau < \tau_{i,l}^{exe'}$  then
4:   Calculate  $\tau_{i,l}^{exe} = \tau_{i,l}^{exe'} - \Delta\tau$ 
5:   Calculate  $t_{i,l}^{end} = t_{i,l}^{end'} - \Delta\tau$  {The end time is re-calculated to start the next interruption period at the correct point in time}
6:   Send a new proposal of  $\tau_{i,l}^{exe}$  to the GRA
7:   Respond to the request of confirmation to the GRA {Agreement for  $\tau_{i,l}^{exe}$ }
8: else
9:   Accept execution period  $\tau_{i,l}^{exe'}$  {Agreement for  $\tau_{i,l}^{exe'}$ }
10: end if

```

4.2 Considering the duration of interruptions

The damping functions $SI(\cdot)$ and $CI(\cdot)$ in Equations (1), (2) determine how much the length of single and cumulative interruptions affect the effectiveness of task execution, and as a result client's utility. The values of $SI(\tau_{i,l}^{int}(t))$ and $CI(\tau_{i,l}^{cum}(t))$ can be calculated at point in time t during task's interruption $\tau_{i,l}^{int}(t)$ to ensure that their values have not substantially dropped towards zero. Assume that a client determines the *sensitivity thresholds* χ^{int} and χ^{cum} with respect to a single and cumulative interruptions respectively. When the difference between the value of any damping function and its largest possible value 1 at time t becomes larger than χ^{int} for $SI(\tau_{i,l}^{int}(t))$ and/or χ^{cum} for $CI(\tau_{i,l}^{cum}(t))$, a client infers that the interruption is too long. An interruption is too long if it significantly affects the client's utility.

Formally, χ^{int} or χ^{cum} denotes the difference between the largest possible value 1 of the corresponding damping function (no impact on client utility) and the values of these functions when the durations of the current single $\tau_{i,l}^{int}(t)$ and/or cumulative $\tau_{i,l}^{cum}(t)$ interruptions exceed the durations $\tau_{int}^{max} - m \times \epsilon_{int}$ and $\tau_{cum}^{max} - k \times \epsilon_{cum}$, as shown in Equations (8), (9).

$$\chi^{int} = 1 - SI(\tau_{int}^{max} - m \times \epsilon_{int}), \quad (8)$$

$$\chi^{cum} = 1 - CI(\tau_{cum}^{max} - k \times \epsilon_{cum}), \quad (9)$$

where the non-dimensional coefficients $m, k \in \mathbb{R}$ denote the amount of the values of ϵ_{int} or ϵ_{cum} which are deducted from τ_{int}^{max} or τ_{cum}^{max} to indicate the start of the time range where the client utility is significantly affected. Considering χ^{int} and χ^{cum} are chosen by a client, we can calculate the value of m from Equations (1) and (8) as in Equation (10) and similarly the value of k from Equations (2) and (9) as in Equation (11).

$$m = \ln \left((1 - \chi^{int}) / \chi^{int} \right), \quad (10)$$

$$k = \ln \left((1 - \chi^{cum}) / \chi^{cum} \right). \quad (11)$$

Here, we describe our additional criteria $C_{i,l}^1(t)$ and $C_{i,l}^2(t)$ for our extended evaluation function, compared to this function in [16]. The previous version of this function consists of only one criterion $C_{i,r}$ for a client to become more generous, when the risk of resource exhaustion in round r for task i is high before the end of negotiation. This criterion is estimated as $C_{i,r} = \sum_{k=1}^r \alpha_{i,k} + r \times \Delta t / t_{dl}$, where $\alpha_{i,k}$ is a relative change of the GRA's

reservation value in respect of the difference of its initial reservation value and its initial offer to a client at the beginning of short negotiation. This criterion $C_{i,r} < 0$, if resource availability mostly decreases with an average speed $|1/r \sum_{k=1}^r \alpha_{i,k}| > \Delta t/t_{dl}$, which means that resources might become exhausted before the end of negotiation. The new criteria $C_{i,l}^1(t)$ and $C_{i,l}^2(t)$, described in this paper, solve the distinct problem of prolonged interruption durations. Consequently, $C_{i,l}^1(t)$ denotes whether the length of the current single interruption is in the time range where the utility is significantly affected by its duration, i.e. $\tau_{i,l}^{int}(t) > \tau_{int}^{max} - m \times \epsilon_{int}$, while $C_{i,l}^2(t)$ has the same idea for the current cumulative interruption, and they are presented below:

$$C_{i,l}^1(t) = (\tau_{int}^{max} - m \times \epsilon_{int}) - \tau_{i,l}^{int}(t), \quad (12)$$

$$C_{i,l}^2(t) = (\tau_{cum}^{max} - k \times \epsilon_{cum}) - \tau_{i,l}^{cum}(t). \quad (13)$$

Our evaluation function $Q_{i,r,l}(r_x, t)$, which is calculated at every point in time t (negotiation round) for the different levels of client greediness, which lead to different rounds r_x of expected agreements. The maximum of this function for the different expected r_x indicates the smallest or largest level of greediness among all levels, produced by the different fuzzy sets' combinations [16]. Therefore, this function consists of the two summands, where only one summand at time t can have a non-zero value: if $r_x \times \Delta t/t_{dl}$ is maximised for the different expected r_x , the longest negotiation is chosen (the largest level of greediness); if $(t_{dl} - r_x \times \Delta t)/t_{dl}$ is maximised, then the shortest negotiation is chosen (the smallest greediness) within a negotiation deadline t_{dl} . A zero or non-zero value of a summand is determined by the Heaviside step functions $\theta(\cdot)$ with criteria as their arguments. For example, if $C_{i,l}^1(t) < 0$, then $\theta(C_{i,l}^1(t)) = 0$ and if $C_{i,l}^1(t) \geq 0$, then $\theta(C_{i,l}^1(t)) = 1$. Our extended evaluation function is below.

$$Q_{i,r,l}(r_x, t) = \left(\frac{r_x \times \Delta t}{t_{dl}} \right) \theta(C_{i,r}) \theta(C_{i,l}^1(t)) \theta(C_{i,l}^2(t)) + \left(\frac{t_{dl} - r_x \times \Delta t}{t_{dl}} \right) \left(1 - \theta(C_{i,r}) \theta(C_{i,l}^1(t)) \theta(C_{i,l}^2(t)) \right). \quad (14)$$

Here, the maximised value of this function over the different combinations of fuzzy sets indicates the largest level of greediness only if $C_{i,r} \geq 0$, $C_{i,l}^1(t) \geq 0$ and $C_{i,l}^2(t) \geq 0$. If at least one criterion is negative, at least one criterion is unfavourable for negotiation, and then a client tends to concede more generously. The flexibility of this evaluation function in terms of adding new criteria for a client to take into account is clear when the new criteria are embedded.

5 Evaluation

We evaluate improvements in client utility compared to our previous negotiation strategy [16], while applying the ConTask negotiation strategy in realistic settings for different Grid environment conditions, reflected in the GRA's reservation value, which changes pseudo-periodically, following changes in resource availability. The GRA's reservation value has also random deviations, which are generated with the normal distribution, representing the more or less predictable GRA's behaviour. As long as deviations in the GRA's reservation value may significantly affect negotiation, we consider them important. That is, we compare client utilities for cases when the standard deviation of the GRA's reservation value can

be 1, 5, 10, 20 and 50 % of the client's optimal value. Larger deviations also cause larger uncertainty for a client in terms of identifying a favourable time interval for negotiation.

We also assume that each short negotiation takes a maximum 100 rounds, equivalent to 100 conventional seconds (but these negotiations may repeat until agreement is reached between a client and the GRA). In our model, all time units are modelled in conventional units, not real time. The duration of execution can be up to 10 days as a result of negotiation, while the total execution period τ^{tot} for each task is equal to 1 conventional year. The period of the change in the GRA's reservation value is considered to be 24 conventional hours.

Client utilities are averaged for 100 tasks, assumed to be executed independently. The variables for the functions $SI(\cdot)$ and $CI(\cdot)$ are considered to be $\tau_{int}^{max} = 100.0$, $\tau_{cum}^{max} = 4800.0$, $\epsilon_{int} = 30.0$ and $\epsilon_{cum} = 2000.0$ conventional seconds. These variables are chosen experimentally, and they generally aim not to significantly affect utility, if a client negotiates successfully during 100 rounds without negotiation repetitions. More repetitions and, as a result, longer interruption periods, affect client utility more significantly.

5.1 Model of resource dynamism

Here, we describe the main principles of our Grid resource dynamism model, which are based on real-life observation of resource utilisation in a Grid, presented in the paper of Iosup et al. [20]. We have identified several pseudo-periodic patterns of resource utilisation fluctuation over time, which are used in this model. Some Grids show a pseudo-periodicity in resource consumption over the days, where period is approximately equal to one day. Although some Grids demonstrate approximately equal peaks of resource consumption every day, others generally show the smaller peaks for the beginning and the end of the week. Some Grids generally show the repetitions in resource utilisation over the days of each week, which we consider as a pseudo-periodicity over the weeks. For example, the resource consumption is generally smaller at the beginning of the week and at the weekends than during the working days. However, some weeks are less distinctive from each other, that is the resource consumption does not drop significantly during the weekends. Some of Grids also do not demonstrate a pseudo-periodic resource utilisation, but they are out of the scope of our research.

Our model is implemented as a simulator, which calculates the amount of free $N_f(t)$ and busy $N_b(t)$ resources, the total amount of resources $N_{tot}(t)$ and the accumulated amount of demanded resources $N_d(t)$ at each time step. This simulator allows us to test the ConTask negotiation strategy for the different scenarios. Figure 3 shows three conventional weeks of resource dynamism simulation, where the total amount of resources is equal to the sum of free and busy resources. In this figure, some days or weeks periodicities in resource consumption are less distinctive than others. For example, the amount of busy resources does not decrease substantially for the weekends between the first and the second weeks (i.e. in the time interval of [120, 168] hours), but it decreases more deterministically for the weekends between the second and the third weeks (i.e. in the time interval of [228, 336] hours). The total amount of resources also changes over time as resources can join or leave a Grid, or fail. The larger resource scarcity is modelled for the cases when the demand on resources increases and / or the total resource amount decreases. Note that the demand on resources $N_d(t)$, depicted in this figure, shows the accumulated amount of requested resources at each time step which has not been allocated yet. In the cases, when the value of $N_d(t)$ close to zero means that there are almost no outstanding resource requests, but it does not mean that there are no new resource requests which is discussed later in this section. In

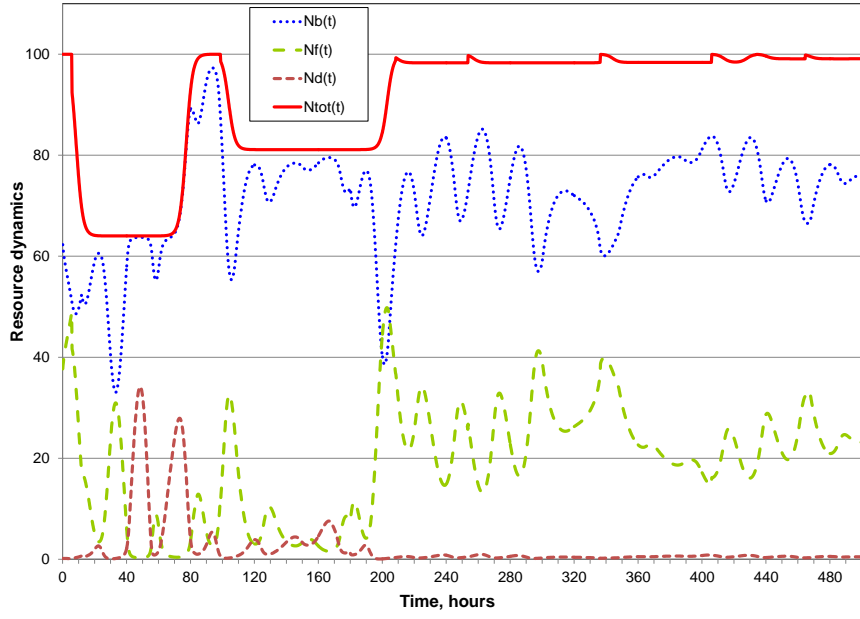


Fig. 3 A demonstration of resource dynamism simulation

order to calculate the amount of free and busy resources as well as the demand on resources at any time point, the system of differential equations is solved numerically:

$$\begin{cases} \frac{dN_f(t)}{dt} = -\alpha \times N_d(t) \times N_f(t) + \beta \times N_b(t) + \mu(t) \times \frac{dN_{tot}(t)}{dt}, \\ \frac{dN_d(t)}{dt} = -\alpha \times N_d(t) \times N_f(t) + \gamma(t) - v(t) \times \frac{dN_{tot}(t)}{dt}, \\ N_{tot}(t) = N_b(t) + N_f(t), \end{cases} \quad (15)$$

where $\alpha \times N_d(t) \times N_f(t)$ or $\beta \times N_b(t)$ are the amounts of resources which become busy or free during an arbitrary small time unit, and α ($1/[\text{time unit}] \times [\text{resource unit}]$) and β ($1/[\text{time unit}]$) are experimentally chosen coefficients.

The change in the total amount of resources $N_{tot}(t)$ also affects the amount of free and busy resources and, as a result, the accumulated amount of requested resources. Fluctuation of $N_{tot}(t)$ over time is simulated using different random superpositions of step-type functions. Here, $\mu(t)$ or $v(t)$ denote the portion of change in the total resource amount for an arbitrary small time unit which are free or busy resources respectively, and $\mu(t) + v(t) = 1$. When the total resource amount increases $dN_{tot}(t)/dt > 0$, then $v(t) = 0$ and $\mu(t) = 1$, i.e. the amount of $N_{tot}(t)$ increases only as a result of additional free resources joining a Grid. When the total resource amount decreases $dN_{tot}(t)/dt < 0$, then $v(t) \in [0, 1]$ and $\mu(t) \in [0, 1]$, i.e. the amount of $N_{tot}(t)$ decreases in respect of free and busy resources as a result of resource failure and / or withdrawal from a Grid.

The accumulated amount of requested resources $N_d(t)$ also depends on the newly requested resource amounts $\gamma(t)$ per time unit, which exhibit periodic dependencies discussed at the beginning of this section. It is assumed that there are more resource requests during

working days than at the weekends, and during the normal working hours than at night. Those periodicities are presented below:

$$\gamma(t) = \gamma_0 \left(1 - \gamma_1 \times \sin\left(\frac{2\pi t}{T_{dyn}}\right) \right)^{d_1} \times \left(\gamma_2 + \left| \sin\left(\frac{2\pi t}{14 \times T_{dyn}}\right) \right|^{d_2} \right), \quad (16)$$

where γ_0 , γ_1 and γ_2 are experimentally chosen constants, i.e. $\gamma_0 = 11$ ([resource unit]/[time unit]), $\gamma_1 = 0.9$ and $\gamma_2 = 0.1$; T_{dyn} is the period of a sine-type wave and is equal to 24 virtual hours; d_1 and $d_2 \in [0, 1]$ are the degrees which define the level of distinction between the peaks of newly requested resource amounts over days and weeks, and t is the current time. The first factor of $\gamma(t)$ models the periodicity in the requested resource amounts over days, while the second factor models the periodicity over weeks. The degrees d_1 and d_2 can be re-calculated randomly in the interval $[0, 1]$ once every 24 virtual hours. The smaller their values, the less distinctive peaks are in resource utilisation over days and weeks. For example, the same level of resource utilisation can be observed for two days without distinctive peaks for each day.

The GRA's reservation value $\tau_{i,r}^{GMx}$ changes proportionally to the amount of free resources $N_f(t)$ with random deviations above mentioned, while the GRA's level of greediness $\beta_{i,r}^G$ changes, depending on the change in the GRA's reservation value and the demand on resources $N_d(t)$ as discussed in Section 3.2.

5.2 Evaluation results

Here, we evaluate each of the contributions of the ConTask negotiation strategy separately and together. In Figure 4, we compare utilities for continuous task execution when a client shortens the best offered execution period “wShort” or a negotiation finishes in a traditional way “noShort”, as explained in Subsection 4.1. In the case of “noShort”, our previous negotiation strategy [16] is used by a client, where the evaluation function has only one criterion $C_{i,r}$ (see Subsection 4.2). The results show that the shortening algorithm improves client utility for all GRA reservation value deviations. However, the difference between utilities in “wShort” and “noShort” becomes gradually smaller towards larger intervals of deviations, due to the increasing uncertainty in estimations of maximum resource availability when the GRA's reservation value has significant oscillations. All utilities decrease towards the higher deviation intervals, because the higher level of deviation of the GRA's reservation value denotes that resources may be exhausted during one or few negotiation rounds when a client has no time to adapt to the resource availability changes.

These results are explained in Figure 5, which shows the execution periods averaged over 100 tasks per half an hour for one day for the different deviation intervals of the GRA's reservation value, where the standard deviation is chosen to be 1% (i.e. “AverExec_1”), 5%, 10%, 20% and 50% of the client's optimal value. As depicted in this figure, smaller deviations (e.g. “AverExec_1” and “AverExec_5”) lead to more accurate estimation of maximum resource availability based on the average execution periods, where this maximum is more distinctive than for larger deviations. However, the longest execution periods around the minimum of resource availability are shown for the largest deviation (i.e. “AverExec_50”), which is due to the drastic increases and decreases in the GRA's reservation value and, as a result, in the durations of allocated time slots of resource utilisation for each task. Consequently, the execution periods on average at the minimum resource availability become almost equal to their values at the maximum of resource availability.

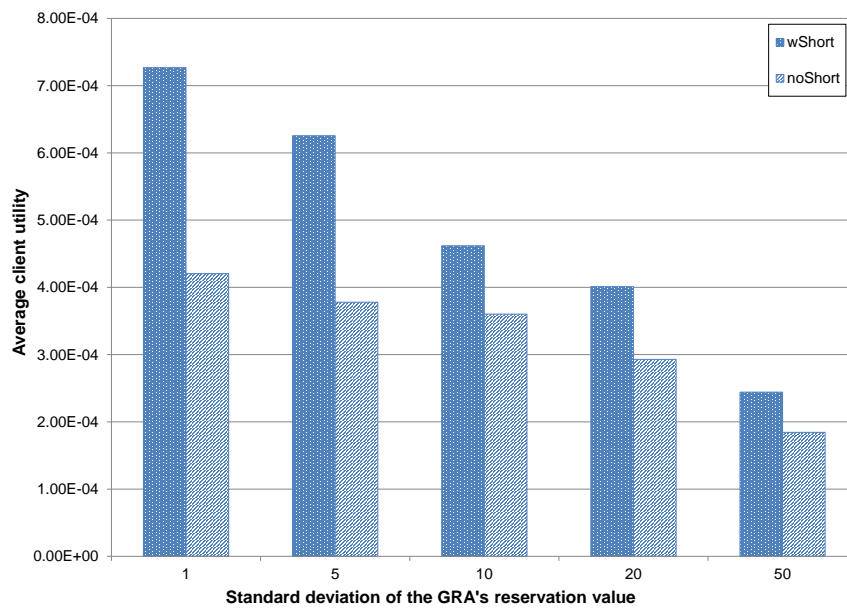


Fig. 4 Evaluation of the client's shortening algorithm (see Algorithm 1)

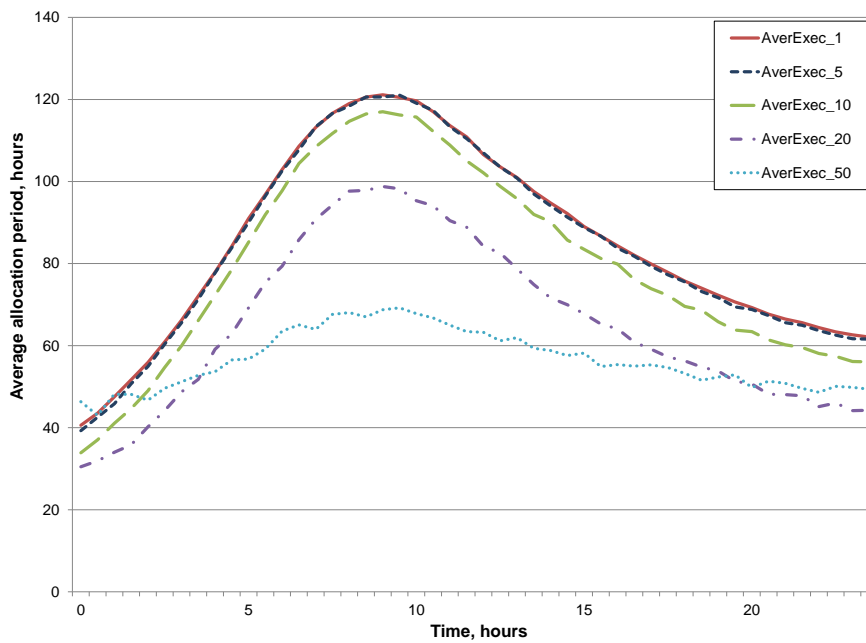


Fig. 5 The average execution periods during one day

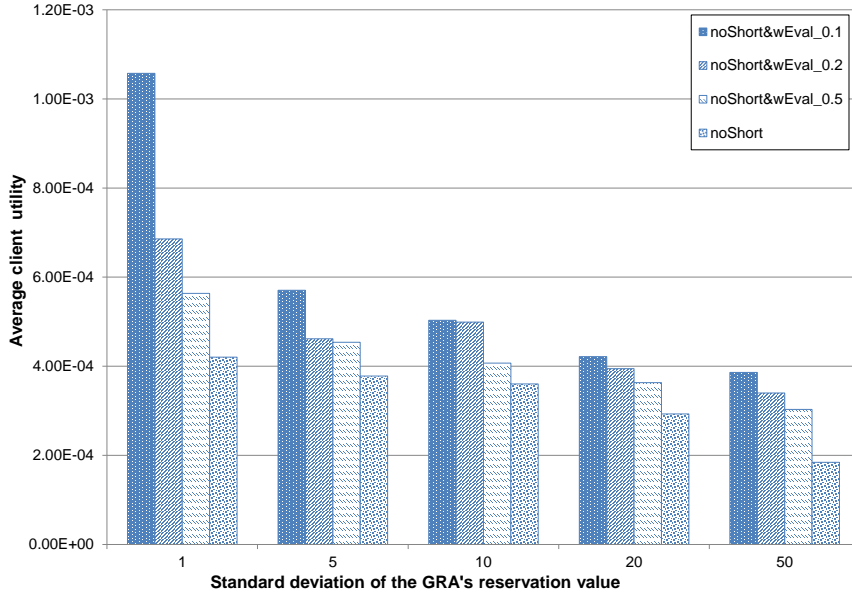


Fig. 6 Evaluation of the client's extended evaluation function

In Figure 6, we compare utilities for our extended evaluation function $Q_{i,r,l}(r_x, t)$ (i.e. “wEval”) with utilities for the case “noShort”. We also evaluate the results for different levels of sensitivity $\chi^{int} = \chi^{tot}$ (see Subsection 4.2), imposed by a client. In this case, we have chosen 0.1 (“noShort&wEval_0.1”), 0.2 (“noShort&wEval_0.2”) and 0.5 (“noShort&wEval_0.5”) for evaluation, because 0.1 represents a mostly generous client, while 0.5 represents a mostly greedy client in negotiation (a client becomes more generous only after the inflection points τ_{int}^{max} and τ_{cum}^{max}). Consequently, 0.2 is chosen as some “medium” level of sensitivity between 0.1 and 0.5.

Figure 6 shows that the smallest utility in all cases is when a client uses our previous non-extended evaluation function “noShort” [16], which does not consider the durations of single and total interruptions. This figure also demonstrates that the smaller sensitivity thresholds, e.g. 0.1, lead to the higher utilities than the larger thresholds. In the cases of the larger sensitivity thresholds, clients greediness leads to longer interruptions when a client does not necessarily negotiate at the maximum resource availability and, as a result, the smaller utilities. It has to be noted that the difference between utilities for all cases decreases with the larger intervals of deviation. This can be explained that the large deviations may as much increase interruption or execution periods at any resource availability as decrease it, because of drastic random increases or decreases in the GRAs reservation value. As a result, the choice of the sensitivity threshold less affects the outcome of negotiation.

However, the durations of interruptions on average for the larger deviations are longer when resources are more scarce, as depicted in Figure 7 (e.g. for the cases with standard deviation 5% (“AvrInt_5”) and 50% (“AvrInt_50”).). This tendency is due to the large negative deviations in the GRA's reservation value which lead to negotiation failures. Note that the average interruptions for the larger deviations around the maximum of resource availability are in turn shorter than for the smaller deviations due to the large negative deviations which

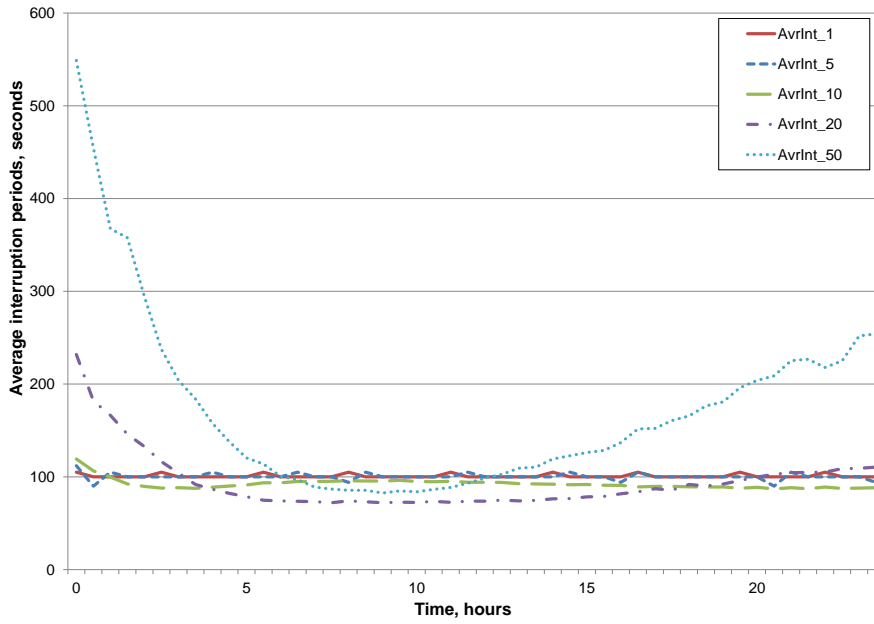


Fig. 7 The average interruption periods during one day, corresponding to the average execution periods in Figure 5

might be perceived by a client as a risk of resource exhaustion and lead to more generous tactics, finishing negotiation earlier. It has to be remembered that the maximal duration of a single negotiation is 100 seconds in our experiment, i.e. if the duration of interruption is close to 100 seconds, it means that the majority of negotiations are successful at first attempt. The average interruption durations for the case “AvrInt_50” at the maximum of resource availability are longer than for the case “AvrInt_20”, which just shows an unpredictable nature of the GRA’s behaviour in the case “AvrInt_50”. That is, the large negative or positive deviations may lead to either too generous (shorter negotiation) or too greedy (longer negotiation) client’s behaviour respectively, averaging closer to the maximal possible duration of negotiation.

In Figure 8, we compare our ConTask negotiation strategy that uses both the shortening algorithm and an extended evaluation function (“wShort&wEval”) to the case when we do not use any of these contributions “noShort”, and generally to all other figures. Here, all utilities are larger than for each separate contribution, as in Figures 4 and 6. It is also important to indicate that the sensitivity threshold 0.2 shows the best utilities for all considered deviations of the GRAs reservation value, which is different to the best sensitivity threshold presented in Figure 6. A less generous client’s behaviour (sensitivity 0.2) becomes more beneficial when each negotiation is likely to start at the maximum of resource availability, while more generous client’s behaviour is beneficial when each negotiation is likely to proceed under unfavourable conditions.

Here, the sensitivity 0.5 leads to the better utilities than 0.1, when the standard deviations are 5%, 10% and 20%, but it shows the worse utilities than 0.1, when the standard deviations are 1% and 50%. It means that a substantial greediness is better than a substantial generosity for almost all deviations, except for the extremely small or large deviations, but it is worse

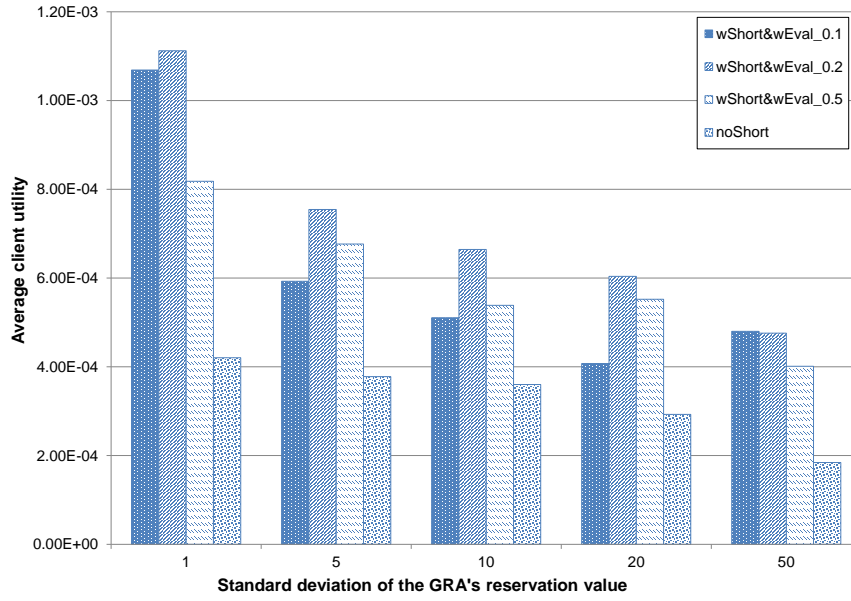


Fig. 8 Evaluation of the whole ConTask negotiation strategy

than a moderate generosity 0.2. In the case of extremely small deviations a greedy client loses in utility in terms of the duration of negotiation for no larger outcome, and in the case of extremely large deviations a greedy client loses in utility, because it is difficult to reach an agreement. A conclusion is that a substantial generosity might be more beneficial for the extreme small or large deviations of the GRA's reservation value, compared to the substantial greediness, while a moderate generosity is better for all cases in general. However, an optimal generosity cannot be chosen precisely and it has to be derived through experimentation for the different Grid environment settings, which can be our further direction of research.

6 Conclusion

In this paper, we described in detail a new negotiation strategy, ConTask, that allows near-continuous task execution by increasing the durations of execution without interruptions and by reducing the interruptions as well. Our strategy offers a different way of reaching agreement where a client may concede beyond the proposal which would be accepted in a traditional negotiation in order to negotiate around the maximum of resource availability. A client also takes into account the durations of the single and cumulative interruptions within the long period of task execution to change its negotiation tactic.

As a result, our negotiation strategy shows an improvement in client utility for different scenarios within a realistically modelled Grid environment, where resource availability changes near-periodically over time. The evaluation results also demonstrate that an extremely generous client's behaviour is more beneficial for the cases when it may not negotiate at the maximum resource availability. A moderately generous behaviour for a client is generally beneficial for the different Grid environments, when a client considers the shorter

execution periods in order to negotiate closer to the maximum of resource availability. It has to be noted that the larger deviations of the GRA's reservation values may lead to the shorter interruptions due to the possible large positive deviations. However, the larger deviations also denote a less predictable Grid's behaviour, which generally leads to smaller client utilities for all strategies.

In our future work, we plan to focus on continuous long-term task execution, considering the dependencies among them in terms of input data streams. In such scenario, an interruption of one task would not just affect its own utility, but also the utilities of other tasks, imposing more time constraints on negotiation process and increasing a complexity of decision-making for a client.

References

1. S. Adabi, A. M. Movaghar, A. Rahmani, and H. Beigy. Negotiation strategies considering market, time and behavior functions for resource allocation in computational grid. *The Journal of Supercomputing*, 66(3):1350–1389, 2013.
2. M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1724–1729, 2006.
3. S. Akioka and Y. Muraoka. Extended forecast of CPU and network load on computational Grid. In *Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid*, CCGrid 2004, pages 765–772, April 2004.
4. A. Andrzejak and M. Ceyran. Characterizing and predicting resource demand by periodicity mining. *The Journal of Network and Systems Management*, 13(2):175–196, 2005.
5. T. Baarslag, K. Fujita, E. H. Gerding, K. Hindriks, T. Ito, N. R. Jennings, C. M. Jonker, S. Kraus, R. Lin, V. Robu, and C. R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198:73 – 103, 2013.
6. T. Baarslag, K. Hindriks, C. Jonker, S. Kraus, and R. Lin. The first automated negotiating agents competition (ANAC 2010). In T. Ito, M. Zhang, V. Robu, S. Fatima, and T. Matsuo, editors, *New Trends in Agent-Based Complex Automated Negotiations*, volume 383 of *Studies in Computational Intelligence*, pages 113–135. Springer Berlin, Heidelberg, 2012.
7. A. Bahrammirzaee, A. Chohra, and K. Madani. An adaptive approach for decision making tactics in automated negotiation. *Applied Intelligence*, 39(3):583–606, 2013.
8. F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov. Adaptive computing on the grid using apples. *IEEE Transactions on Parallel and Distributed Systems*, 14(4):369 – 382, 2003.
9. P. M. Castro, A. P. Barbosa-Póvoa, H. A. Matos, and A. Q. Novais. Simple continuous-time formulation for short-term scheduling of batch and continuous processes. *Industrial & Engineering Chemistry Research*, 43(1):105–118, 2004.
10. L. Chen, K. Reddy, and G. Agrawal. GATES: a grid-based middleware for processing distributed data streams. In *Proceedings of the 13th IEEE International Symposium on High performance Distributed Computing*, pages 192–201, 2004.
11. P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159 – 182, 1998.
12. I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15:200–222, 2001.
13. M. Ghanem, Y. Guo, J. Hassard, M. Osmond, and M. Richards. Sensor grids for air pollution monitoring. In *Proceeding of the 3rd UK e-Science All Hands Meeting*, 2004.
14. V. Haberland, S. Miles, and M. Luck. Adaptive negotiation for resource intensive tasks in Grids. In K. Kersting and M. Toussaint, editors, *Proceedings of the 6th Starting AI Researchers' Symposium*, volume 241 of *Frontiers in Artificial Intelligence and Applications*, pages 125–136. IOS Press, 2012.
15. V. Haberland, S. Miles, and M. Luck. Negotiation to execute continuous long-term tasks. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1019–1020. IOS Press, 2014.
16. V. Haberland, S. Miles, and M. Luck. Adjustable fuzzy inference for adaptive grid resource negotiation. In K. Fujita, T. Ito, M. Zhang, and V. Robu, editors, *Next Frontier in Agent-based Complex Automated Negotiation*, volume 596 of *Studies of Computational Intelligence*, pages 37–57. Springer Japan, 2015.

17. K. Hindriks, C. M. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: Negotiation environment for heterogeneous agents. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of AAMAS '09, pages 1397–1398, Richland, SC, 2009.
18. K. Hindriks, C. M. Jonker, and D. Tykhonov. The benefits of opponent models in negotiation. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, volume 2, pages 439–444, 2009.
19. C. Hou. Predicting agents tactics in automated negotiation. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004.*, pages 127 – 133, 2004.
20. A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. J. Epema. The grid workloads archive. *The Journal of Future Generation Computer System*, 24(7):672–686, 2008.
21. D. Kondo, M. Taufer, C. Brooks, H. Casanova, and A. Chien. Characterizing and evaluating desktop grids: an empirical study. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, pages n/a–n/a, 2004.
22. F. Lang. Developing dynamic strategies for multi-issue automated contracting in the agent based commercial grid. In *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, volume 1, pages 342 –349, 2005.
23. E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, C-26(12):1182 –1191, 1977.
24. V. Narayanan and N. R. Jennings. An adaptive bilateral negotiation model for e-commerce settings. In *Proceedings of the 7th International IEEE Conference on E-Commerce Technology*, pages 34–39, 2005.
25. G. R. Nudd, D. J. Kerbyson, E. Papaefstathiou, S. C. Perry, J. S. Harper, and D. V. Wilcox. PACE—a toolset for the performance prediction of parallel and distributed systems. *High Performance Computing Applications*, 14(3):228–251, 2000.
26. F. Ren, M. Zhang, and K. M. Sim. Adaptive conceding strategies for automated trading agents in dynamic, open markets. *Decision Support Systems*, 46(3):704–716, 2009.
27. A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.
28. G. C. Silaghi, L. D. Șerban, and C. M. Litan. A time-constrained SLA negotiation strategy in competitive computational grids. *Future Generation Computer Systems*, 28(8):1303–1315, 2012.
29. K. M. Sim. Equilibria, prudent compromises, and the “waiting” game. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(4):712 –724, 2005.
30. K. M. Sim. From market-driven e-negotiation to market-driven g-negotiation. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 408–413. IEEE Computer Society, 2005.
31. K. M. Sim. Evolving fuzzy rules for relaxed-criteria negotiation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(6):1486–1500, 2008.
32. K. M. Sim and K. F. Ng. A relaxed-criteria bargaining protocol for grid resource management. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, volume 2, pages 8 pp.–5, 2006.
33. K. M. Sim and S. Y. Wang. Flexible negotiation agent with relaxed decision rules. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(3):1602–1608, 2004.
34. D. Spooner, S. Jarvis, J. Cao, S. Saini, and G. Nudd. Local grid scheduling techniques using performance prediction. *IEEE Proceedings - Computers and Digital Techniques*, 150(2):87 – 96, 2003.
35. R. Wolski. Experiences with predicting resource performance on-line in computational grid settings. *SIGMETRICS Performance Evaluation Review*, 30(4):41–49, 2003.
36. R. Wolski, N. T. Spring, and J. Hayes. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(5-6):757–768, 1999.
37. M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10:115–152, 1995.
38. Y. Yuan, Y. Wu, G. Yang, and W. Zheng. Adaptive hybrid model for long term load prediction in computational grid. In *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid, CCGRID '08*, pages 340–347, May 2008.
39. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.